



Bottom-up, Open, and Concurrent

A Perspective on Development Processes for Scientific Software Systems

Jing Guo

SAIC/General Sciences Corporation, Laurel, Maryland

and

P. M. Lyster

Department of Meteorology and Earth System Science Interdisciplinary Center

University of Maryland, College Park, Maryland

25th Annual Software Engineering Workshop, November 28-30, 2000

NASA/Goddard Space Flight Center, Greenbelt, Maryland

Outline

1. About the PSAS development
2. Characteristics of the PSAS development process:
 - Bottom-up
 - Open
 - Concurrent
3. Summary

About the PSAS Development

- The Physical-space Statistical Analysis System (PSAS) is a complex scientific computing system, both computational and data intensive (≈ 90 KLOC). However, it is still just one of several complex subsystem components for the DAO's Data Assimilation System.
- It was developed over a ten year period, involving a total of less than a dozen developers (≈ 20 man-years), and continuously supported scientific research efforts and the DAO's data assimilation systems.
- Its developers are scientists with advanced degrees and strong scientific computing background, as well as shared interests of delivering quality software and improving the process.

About the PSAS Development (cont.)

- The process was *ad hoc* and “heroic” by software engineer’s standards. It needs to be improved.
- The process was repeatable and understandable among the developers. Should one take a seriously look at the process before “improve” it?

Characteristics of the PSAS development process

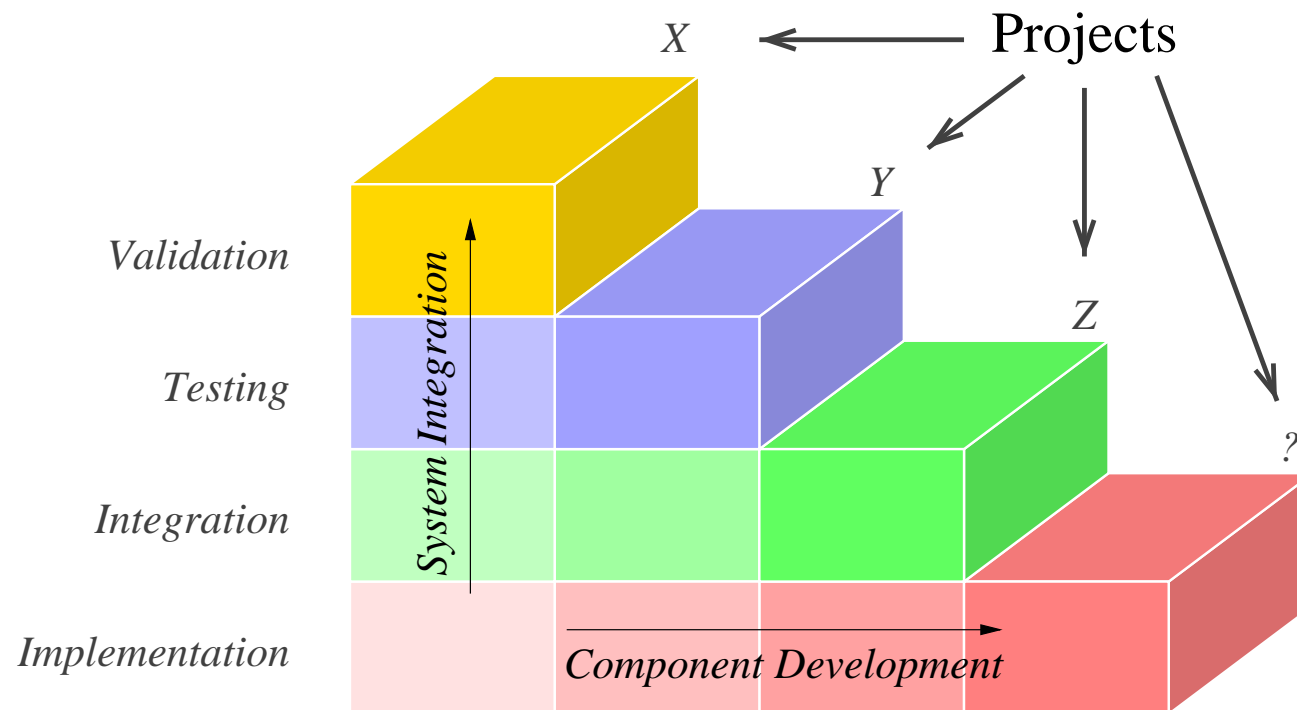
1. Bottom-up
 - get components ready, before plan for an integration
2. Open
 - let users and developers to have the control
3. Concurrent
 - resolve conflicts when and only when they appear

Bottom-Up

Get components ready, before plan for an integration

- Developing components first is a common sense risk mitigation to make a system integration possible.
- Components development must be driven by a strategic view on the future direction of the software, while customizing the releases for different system integration projects.
- Planning system integration projects is very risky without certainties on its components releases.
- Compromise is necessary between the process for long-term sustainable components development, and the process required by short-term projects.

Bottom-Up in a Component Development View



Open

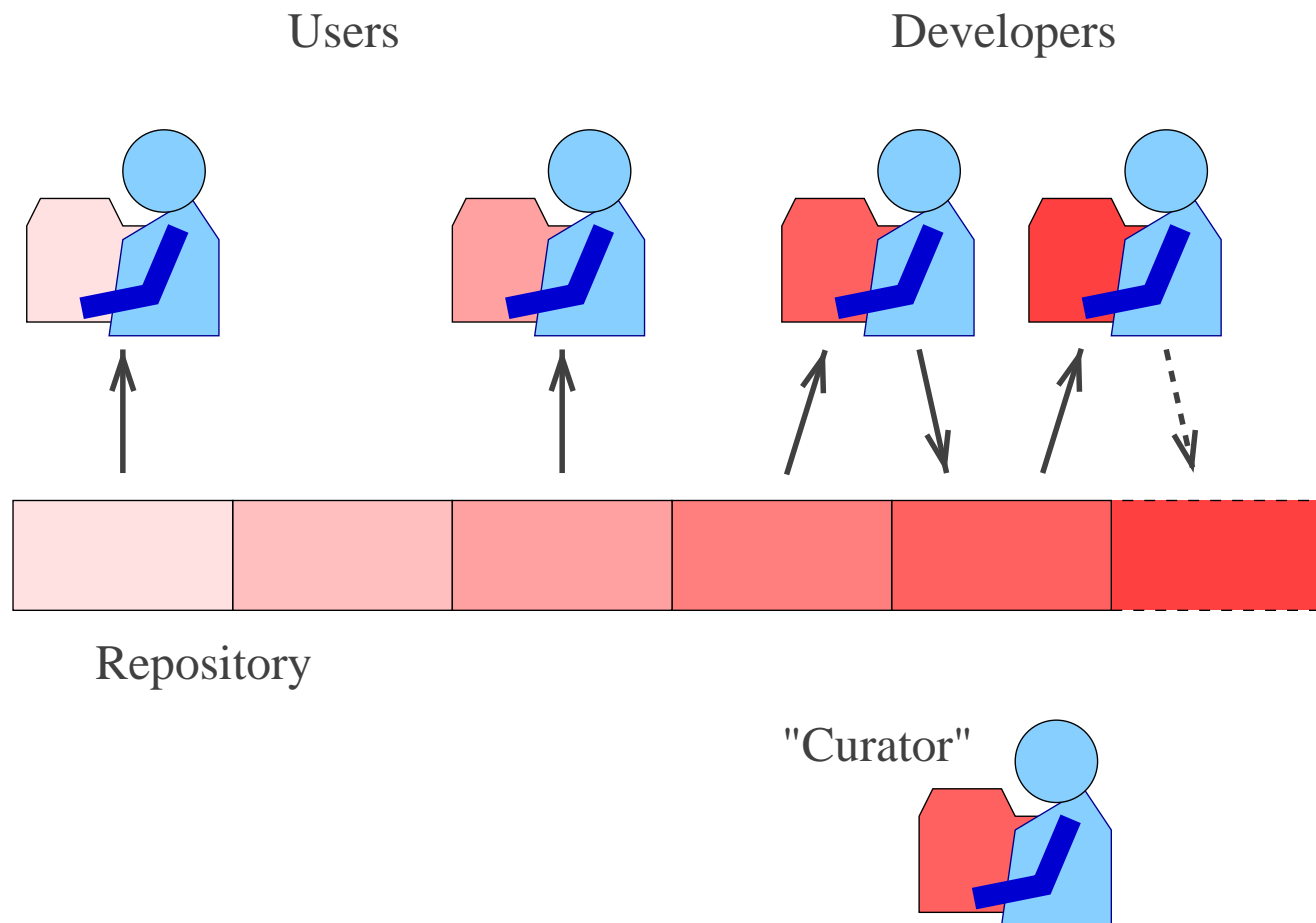
Let users and developers to have the control

- A scientific software system will not be successful without the supports of its users and the involvements of its scientific community.
- Managing the changes to the software made by its users through the openly accessible source code is a constant challenge.
- For the PSAS development, users and developers are allowed, invited, and guided, to make changes, to contribute the changes to the system based on their own best scientific and personal managerial judgment.
- Software releases are safe when software are properly managed, e.g. through a version management system.

Different Roles in an Open Development

- Users (e.g. system integrators) should have the final control over what is to be used based on their requirements.
- Developers often have the best knowledge about the the solutions best fit the requirements to make detailed decisions in the right place and at the right time.
- A curator is needed to balance the choices of the users and the changes implemented by the developers, through releases.

An Open Development Environment

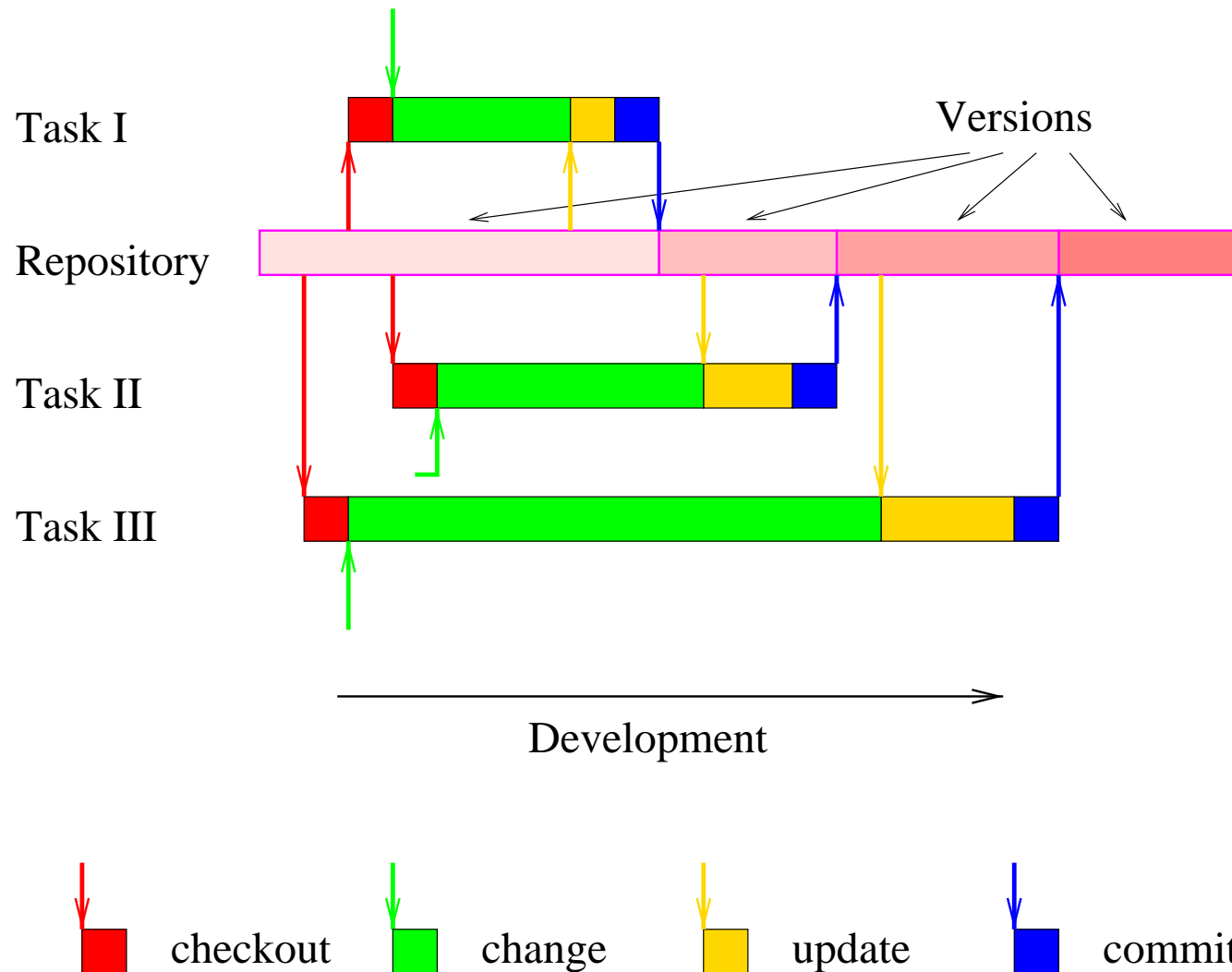


Concurrent

Resolve conflicts when and only when they appear

- Most development activities for a scientific software are already concurrent without being properly managed, resulting conflicts between development activities.
- The more open the development is, the more concurrent these development activities become.
- Resolving conflicts when and only when they occur is efficient concurrent development policy, based on an optimistic view of conflicts.

Basic Activities in Concurrent Development



Summary

- Because of the complex, exploratory, and collaborative nature of scientific software development, its process often has the characteristics of being *bottom-up*, *open*, and *concurrent*.
- These characteristics are important to encourage innovations in scientific software development, and to ensure sustainable long-term system developments.
- These characteristics may present challenges to the management of similar developments. However, processes with these characteristics may be implemented by working around the actual practices.